Colin Pham

Professor Brooks

Foundations of AI

23 September 2024

Assignment 2: Search - Question 5 - Deep Blue vs Alpha Zero

Deep Blue was an artificial intelligence that was successful in beating a grandmaster level chess player at their own game. Its success mainly stems from the fact that it's a culmination of learned mistakes and improvement from previous iterations. Such improvements revolved around the chips it runs on. Because of Deep Blue's 216 chess chips, it was able to visualize 2.5 million chess moves and calculate an optimal move given a scenario. Its next iteration, Deep Blue II, featured not only twice the amount of chips, but also a way to remove duplicate moves, thus enhancing its calculation speeds. How the AI actually worked with the data was the one thing that was focused on. In the end, game specific tuning was the cause of its victory against Garry Kasparov. Other problems that require the knowledge of every possible combination can be used with the kind of chips Deep Blue utilized. Also, since it favors a tree-like structure and search, any sequence based problem (or any problem that can be looked at as one) is compatible with these chips. The real problem, the evaluation step, stems from learning from past mistakes. Thus, previous chess games can only benefit chess specifically while previous problem attempts can only benefit said problems.

Alpha Zero and Stockfish were put head to head in chess with Alpha Zero winning all of its games despite having 1/10 of Stockfish's calculation time. This is all due to its method of calculation. Though it uses Monte Carlo to play every single possible combination, it calculates the probability of a desired outcome rather than a win. Trying to calculate every single chess

possibility at once is long. Dividing the problem up into smaller ones and playing to get to certain steps is more optimal and leaves room to account for deviations. To frame it in a simpler light, think about a chess game with a starting state and a winning state. A victorious chess game consists of different steps, or nodes. Say there are multiple scenarios of a chess game, and nodes a, b, and c are the outcomes needed in order to win said scenarios. Calculating every possible scenario and steps would be cumbersome, especially during the start state where anything can happen. Instead, start from the beginning and work on node a, in which a more concise strategy can be calculated more easily depending on the opponent's move so far. Doing the same for nodes b and c would result in a faster calculation.